UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/608,313 | 06/30/2000 | Gerolf F. Hoflehner | 042390.P8132 | 9388 |

7590 05/06/2004

Robert B O'Rourke
Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026

| EXAMINER | |
|---|---|
| SHRADER, LAWRENCE J | |
| **ART UNIT** | **PAPER NUMBER** |
| 2124 | 13 |

DATE MAILED: 05/06/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. 09/608,313 | Applicant(s) HOFLEHNER ET AL. | |
|---|---|---|---|
| | Examiner Lawrence Shrader | Art Unit 2124 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on 2/23/2004;3/08/2004.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) 1-80 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-38, 40 - 78, and 80 is/are rejected.

7)☒ Claim(s) 39 and 79 is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

1. This office action is in response to the amendment filed on 2/23/2004 by the Applicant.

### *Drawings*

2. The formal drawings filed on 3/08/2004 are acknowledged.

3. Claims 1 – 38, 40 – 78, and 80 remain rejected.

4. Applicant's arguments with respect to claims 1 and 14 have been considered but are moot in view of the new ground(s) of rejection.

### *Claim Rejections - 35 USC § 102*

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1, 28 – 34, 37, 40; 68 – 74, 77, and 80 are rejected under 35 U.S.C. 102(b) as being anticipated by Wu, U.S. Patent 6,230,317.

**In regard to claim 28:**

*"performing a first allocation for a first amount of on-processor register space at the entry block of a routine said first amount of on-processor register space for the use of said routine;"*

Wu teaches allocation of register space at the entry block of a routine to be used (column 3, line 64 to column 4, line 28)

*" performing a second allocation for a second amount of on-processor register space for the storage of live information of said routine when said routine performs a function call to a second routine, said second amount less than said first amount;"*

Wu teaches allocation of a second amount of resister space for live information (the loop routine contains information; column 4, lines 1 – 15).

*" performing said function call to said second routine;"*

The second routine (loop) is called (column 4, lines 11 – 21).

*"performing a third allocation for a third amount of on-processor register space at the entry block of said second routine, said third amount of allocated on-processor register space for the use of said second routine and including a region of said first amount of allocated on-processor register space that stores stale information of said routine when said routine said performs said function call to said second routine."*

Wu teaches a third allocation with an overlapping registers space (see Figure 3 and column 4, lines 14 – 17)

**In regard to claim 29,** incorporating the rejection of claim 28:

*"...said live information is determined by identifying information that is referred to before and after said function call."*

The live data in inherently included based on information referred to before and after the function call.

**In regard to claim 30,** incorporating the rejection of claim 29:

*"...information identified after said function call extends to an exit block of said routine."*

(e.g., see Figure 2b)

**In regard to claim 31,** incorporating the rejection of claim 30:

*"...the worst case path to said exit block is allocated for."* (see figure 2b)

**In regard to claim 32 and 33** incorporating the rejection of claims 29 and 32 respectively:

*"...information identified after said function call extends to a post-dominator block of said routine."*

It is understood that a function call extending to an exit block would inherently apply as well to a post-dominator block since the type of flow graph would not alter the method. *"...the worst case path to said post-dominator block is allocated for."* (see figure 2b)

**In regard to claim 34,** incorporating the rejection of claim 28:

*"...said live information is information that is local to said routine."*

The information in the loop function is live information (Figure 2b).

**In regard to claim 37,** incorporating the rejection of claim 28:

*"...said second allocation is performed just before said function call."*

The second storage area is allocated before the function is called (column 4, lines 9 – 14).

**In regard to claim 40,** incorporating the rejection of claim 28:

*"...compiling said routine."* Compiling or interpreting is an inherent step of preparing a routine to run a computer system.

**As to claims 68 – 74, 77, and 80,** they are rejected for the same reasons as put forth for claims 28 – 34, 37, and 40 above, the claims being a machine-readable medium version of the claimed method discussed and/or addressed above in claims 28 – 34, 37, and 40.

**As to claim 1**, rejected for the same reasons put forth in the rejection of claim 37 above.

### *Claim Rejections - 35 USC § 103*

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1, 11 – 16, 19, 20, 23, and 24; 41, and 51 – 53; 54 – 56, 59, 60, 63, and 64 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, U.S. Patent 5,367,684 in view of Wu, U.S. Patent 6,230,317.

**In reference to claim 1:**

*"inserting an on-processor register allocation instruction within a machine code description of a routine if a function call instruction is found within said routine, said inserted on processor register allocation instruction to allocate less on-processor register space for the use of said routine than an amount of on-processor register space allocated for the use of said routine by another on-processor register allocation instruction that is executed prior to said inserted on processor register allocation instruction."*

The allocation instruction in claim 37 is inherently present in order to guide the allocation of memory. However, assuming it is not, then the insertion of an allocation instruction follows according to claim 1. Smith discloses a register allocator, which allocates registers to instructions of each basic block of code (column 7, lines 9 – 15), and inserts spill instructions (column 7, lines 16 – 21) into the block of instructions based on a live range, but does not explicitly teach the identification of a function call as the indicator of a block, although one skilled in the art could infer that the Smith invention refers to function blocks (e.g., see Figure 7f). Wu, however, explicitly teaches the determination of an instruction in the form of a function call in a code segment (e.g., Figure 2B) in order to allocate storage area. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the function call identification method of Wu with the instruction identification and insertion feature taught by Smith invention, which clearly locates a basic block of code in a program (and a function defines a basic block of code) in order to insert a spill instruction to allocate memory, because one skilled in the art would understand that a function call is an instruction that the Smith invention would recognize as identifying a block of code.

**In reference to claim11**, official notice is taken for allocation of an instruction inserted just before the function-call, because one of ordinary skill would expect that the proper allocation configuration be made *before* the function runs. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to insert an allocation instruction before the function call so that the allocation of resources for the function is complete before the function is run.

**In reference to claims 12 and 13**, official notice is taken for allocation of an instruction inserted before the function-call because one of ordinary skill in the art would expect that the proper allocation configuration be made before the function runs. Also, a pre-dominator block implies a dominator block, which in turn implies that a post-dominator block exists, i.e., a block being dominated. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to insert an allocation instruction before the function call, whether or not in a pre-dominator block, so that the allocation of resources is complete before the function is run.

**In reference to claim 14**, rejected for the same reasons put forth in the rejection of claim 1. Having loops and pipelining and calling to another routine as discloses by Wu in claim 1 would simply multiply the insertion of allocation instructions whenever a functional characteristic is found after searching. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to search for multiple functional characteristics and apply the allocation instruction insertions in order that each functional routine or block of code might contribute to the greater efficiency of the program.

**In reference to claims 15, 19, and 23**, rejected for the same reasons put forth for claim 1. A functional characteristic corresponding to either a loop in a control flow graph, or a software pipelined loop may be interpreted as a routine resulting from a function call. In the case of a loop, a section of code need not be repeated as inline code; in the case of a software pipelined loop, the function may utilized so that the code might be optimized for greater throughput.

**In reference to claims 16 and 20**, rejected for the same reason put forth in the rejection of claim 11.

**In reference to claim 24**, a method for determining the number of registers to be allocated for an allocation instruction inherently follows an analysis of functional characteristics of the routine.

**Claims 41, and 51 – 53** (the medium) are rejected for the same corresponding reasons put forth for claims 1, and 11 – 13 (the method).

**Claims 54 – 56, 59, 60, 63, and 64** (the medium) are rejected for the same corresponding reasons put forth for claims 14 – 16, 19, 20, 23, and 24 (the method).

9. Claims 2, 8 – 10; 42, and 48 – 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, U.S. Patent 5,367,684 in view of Wu, U.S. Patent 6,230,317, as applied to claim 1 above, and further in view of Proebsting et al., "Demand Driven Register Allocation" (hereinafter referred to as Proebsting).

**In reference to claim 2**, Smith discloses allocation instructions, and Wu teaches the determination of a function call in a routine. Neither Smith nor Wu teaches allocation for only live information, but Probsting teaches the allocation of registers for live information (page 683, Abstract). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the disclosures of Smith and Wu in order to insert allocation instructions in a routine if a function call instruction is found and to further modify the combination with the use of live data as taught by Proebsting so that only useful information is stored.

**In reference to claim 8**, Proebsting teaches local values to allocate registers to the given routine (page 1, Abstract) as applied to claim 2.

**In reference to claim 9**, Smith discloses allocation instructions, and Wu teaches the determination of a function call in a routine. Neither Smith nor Wu teaches a register space partitioned for global variables and one for local variables with said instruction allocating local register space. Proebsting teaches the allocation of both local and global registers (page 1, Abstract). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to construct an allocation instruction according to the combination of Smith and Wu and further implement the instruction with the ability to partition register space for local variables, while allocating global registers by another means, as taught by Proebsting.

**In reference to claim 10**, Smith discloses allocation instructions, and Wu teaches the determination of a function call in a routine. Neither Smith nor Wu teaches allocation for live information that is global information, but Proebsting teaches the allocation of registers for global live information (page 1, Abstract). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the function call identification method of Wu with the insertion of allocation instructions taught by Smith in order to insert allocation instructions in a routine if a function call instruction is found, and further modified by Proebsrting to include live variable information that is global so that a single register might be allocated for several variables that are not simultaneously live.

**Claims 42, and 48 - 50** (the medium) are rejected for the same corresponding reasons put forth for claims 2, and 8 – 10 (the method).

10. Claims 3 – 7; 17, 18, 21, 22, 25 – 27; 43 – 47; 57, 58, 61, 62, and 65 – 67 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, U.S. Patent 5,367,684 in view of Wu, U.S. Patent 6,230,317, and further in view of Proebsting et al., "Demand Driven Register Allocation" (hereinafter referred to as Proebsting), as applied to claim 2 above, and further in view of Srivastava, U.S. Patent 5,999,737.

**In reference to claims 3, 4, and 6** Smith discloses allocation instructions, and Wu teaches the determination of a function call in a routine; Proebsting teaches the allocation of live information for both local and global registers. None specifically teach the method of determining live information by identifying information that is referred to before and after a function call (claim 3) and extending to the exit block of the routine (claim 4) or a post-dominator block (claim 6) - it is understood that a function call extending to an exit block would apply as well to a post-dominator block since the type of graph would not alter the method. Srivastava, on the other hand, teaches a liveliness analysis wherein a set of live variables is determined at the beginning of execution of a function call and also at the end of the function call (column 8, lines 6 – 12); the liveliness analysis also extends to the exit block, or a post-dominator, of the routine (10, lines 12 – 26). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to insert an allocation instruction as determined by the combination of the disclosures of Smith and Wu, and configuring the instruction to allocate only live information as modified by Srivastava, where said information is referred to both before and after a function call, and where the identified information extends to

an exit block of the routine also taught by Srivastava so that resources are efficiently allocated for only useful data, and for increasing the efficiency of the compiled program.

**In reference to claims 5 and 7**, official notice is taken that none of the cited references work specifically to eliminate a worst-case path, therefore, one may infer that the worst-case path may be allocated, especially in Probsting and Srivastava.

**In reference to claims 17 and 21**, rejected for the same reason put forth in the rejection of claim 5.

**In reference to claims 18**, rejected for the same reason put forth in the rejection of claim 7.

**In reference to claim 22**, rejected for the same reason put forth in the rejection of claim 6.

**In reference to claims 25 and 26,** rejected in line with the reasoning for claim 24. Functional characteristics must be known before a determination of a consequent action can be made.

**In reference to claim 27,** building an understanding of a flow control graph is inherent in a compiling procedure, therefore, searching could not be done prior to the understanding.

**Claims 43 – 47** (the medium) are rejected for the same corresponding reasons put forth for claims 3 – 7 (the method).

**Claims 57, 58, 61, 62, and 65 – 67** (the medium) are rejected for the same corresponding reasons put forth for claims 17, 18, 21, 22, and 25 – 27 (the method).

11. Claims 35, 36, and 38; 75, 76, and 78 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wu, U.S. Patent 6,230,317, as applied to claim 28 above, in view of Proebsting et al., "Demand Driven Register Allocation" (hereinafter referred to as Proebsting).

**In regard to claim 35,** incorporating the rejection of claim 34:

*"...associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space for local information."* Wu teaches register storage allocation, but does not teach partitioning of register space. However, Proebsting teaches allocation of both local and global registers (page 683, Abstract). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to allocate register space according to the teaching of Wu and further implement the instruction with the ability to partition register space for local variables, while allocating global registers by another means, as taught by Proebsting, because Proebsting teaches that local allocation precedes global allocation, which is done iteratively in nested loops (Abstract), which would apply to the nested loop feature of Wu.

**In regard to claim 36,** incorporating the rejection of claim 28:

*"...live information includes global information."* Wu teaches register storage allocation, but does not specifically teach that live information includes global information. However, Proebsting teaches that global information includes global information (page 683, Abstract).

Therefore, it would have been obvious to one skilled in the art at the time the invention was made to allocate register space with live local information according to the teaching of Wu and further implement the instruction with the ability to include live global information as taught by Proebsting, because Proebsting teaches that local allocation precedes global allocation over live ranges, which is done iteratively in nested loops (Abstract), which would apply to the nested loop feature of Wu.

**In regard to claim 38,** incorporating the rejection of claim 28:

*"...second allocation is performed in a pre-dominator basic block of said function call."*

Wu teaches register storage allocation, but does not teach allocation performed in a pre-dominator block of the function call. However, Proebsting allocation performed in a pre-dominator block (pre-header; e.g., see Figure 10). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to allocate register space according to the teaching of Wu and further implement the ability to allocate in a pre-dominator basic block as taught by Proebsting, because Proebsting teaches that the pre-header is also considered to be a register-definition of every variable since loads can be added there as needed (page 692, first para. of section 3.5).

**As to claims 75, 76, and 78,** they are rejected for the same reasons as put forth for claims 35, 36, and 38 above, the claims being a machine-readable medium version of the claimed method discussed and/or addressed above.

***Allowable Subject Matter***

12. Claim 39 (the method) and claim 79 (the machine readable medium) is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

***Response to Arguments***

13. Applicant's arguments filed on 2/23/2004 have been fully considered but they are not persuasive.

**(A) The Applicant has argued:**

Independent claim 1 presently stands rejected under 35 USC 102(b) as being anticipated by U.S. Patent No. 6,230,317 (hereinafter "Wu") and stands rejected under 35 USC 103(x) as being obvious in light of the combination of U.S. Patent 5,367,684 (hereinafter, "Smith") and U.S. Patent No. 5,748,963 (hereinafter, "Orr"). Independent claim 1 presently recites (emphasis added):

1. A method, comprising:
inserting an on-processor register allocation instruction within a machine code description of a routine if a function call instruction is found within said routine, said inserted on processor register allocation instruction to allocate less on-processor register space for the use of said routine than an amount of on-processor register space allocated for the use of said routine by another on-processor register allocation instruction that is executed prior to said inserted on processor register allocation instruction.

Claim 1 recites a pair of allocation instructions ("an on-processor register allocation instruction" and "another on-processor register allocation instruction") that allocate registers for the use of the same routine. Moreover, claim 1 recites that the later executed allocation instruction allocates for less register space than the earlier executed allocated allocation instruction.

These claimed features stem at least from the Applicants' discussion of Figures 5 and 6 of the present application. That is, Figures 5 and 6 of the present application and their surrounding discussion disclose a technique where an allocation instruction that allocates register space (e.g., space 610 in Figure 6) for a caller routine (as a consequence of the caller routine having a function call) can be configured to allocate for less register space than a previously executed allocation instruction allocated (e.g., space 605 of Figure 6)for the same caller routine (e.g., located at the caller routine's entry block).

By contrast the Wu reference at best only discloses, teaches or suggests an allocation instruction for the called routine rather than the caller routine. As such, the Wu reference fails to cover the Applicants' claimed subject matter. The Wu

reference teaches the implementation of a software pipelined loop as a called sub function and the corresponding allocation of register space for the called sub function. See, Wu Col. 3, lines 26-29 and Col.4 lines 5-21. Nothing is said in Wu with respect to the insertion of an allocation instruction that allocates register space for the caller function, however. Therefore the Wu reference fails to anticipate each and every limitation of independent claim 1 of the present application.

Similarly, the combination of Orr and Smith fail to disclose, teach or suggest a causal relationship between the presence of a function call in a caller routine and the insertion of an additional allocation instruction in the caller routine that allocates register space for the caller routine. As discussed at length in the Office Action response mailed October 15, 2003, Orr teaches a wholly irrelevant causal relationship between the presence of a function call instruction and the replacement of function call parameters with a dictionary entry.

Smith teaches a register allocation method (specifically, the allocation of register space to two "register candidates" that are live within the same basic block of instructions through an improved "register candidate usage matrix". See, Smith, Col. 2, lines 5-32; Col.2, line 60 -Col. 3, line 2) but says nothing about the insertion of an allocation instruction. That is, Smith teaches a method that can perhaps be used to determine "how many" registers are to be allocated for by an allocation instruction; but, by contrast, fails to disclose how many allocation instructions are to be inserted into a routine and/or a function call made from the routine being a stimulus for the insertion of an allocation instruction into the routine.

Because Orr is a wholly irrelevant reference and because Smith fails to disclose any details regarding the insertion of an allocation instruction the combination of Smith and Orr simply fail to cover the matter claimed by independent claim 1 of the present application.

Therefore the Applicants respectfully submit that independent claim 1 and each of its corresponding dependent claims are patentable over the outstanding theories of rejection.

**Examiner's Response:**

Firstly, claim 1 refers to inserting an on-processor register allocation instruction, and the applied art reads on that action. There are no other steps in the claim other than inserting. Simply appending more information onto the claim does not necessarily add any patentable weight to the single step of "inserting an on-processor register allocation instruction." A question arises: Are there any other analyses or steps to determine how much space is less than previously allocated? It is not clear in the claim how this gets done.

Also, the Applicant makes a distinction between a caller routine and a called routine, which is not even mentioned in the claim; therefore it appears to be a moot argument. If this distinguished the invention over the prior art, then that feature should be in the claims.

(B) **The Applicant has argued:**

Independent claim 14 presently stands rejected under 35 USC 103 as being obvious in light of the combination of Smith and Orr. Independent claim 14 presently recites:

> A method comprising: inserting an on-processor register allocation instruction within a machine code description of a routine because a functional characteristic selected from the group consisting of:
>
> a) a loop that exists within a control flow graph of said routine;
>
> b) a software pipelined loop; and,
>
> c) a function call to another routine;
>
> is discovered within said routine, said inserted on-processor register allocation instruction to allocate on processor register space for the use of said routine.

The Applicants respectfully submit that Independent claim 14 is patentable over the combination of Orr and Smith because, as discussed above, Orr is a wholly irrelevant reference and Smith fails to disclose any details regarding the insertion of an allocation instruction. In particular, both Orr and Smith fail to disclose any details regarding a particular functional characteristic (e.g., a loop, a software pipelined loop and/or a function call) being a stimulus for the insertion of an allocation instruction. Because neither of these references individually teach or suggest a specific stimulus for the insertion of an allocation instruction, it is impossible for their combination to disclose, teach or suggest any such stimulus. Better said, the Applicants claim a causal relationship between the presence of a particular functional characteristic within a routine and the insertion of an allocation instruction within the routine. Because there is simply no relationship between Orr and Smith, the "function call" of Orr cannot be related to (what is at most a suggestion of the existence of) an allocation instruction as provided by Smith. Therefore independent claim 14 and its corresponding dependent claims are patentable over the combination of Orr and Smith.

**Examiner's response:**

The rejection of claim 14 was based on looping, which would simply multiply the insertion of allocation instructions whenever a functional characteristic is found after searching as covered in claim 1, and a function call was also rejected in claim 1 with the combination of Smith and Wu wherein a call to another routine is made.

## (C) The Applicant has argued:

Independent claim 28 presently stands rejected under 35 USC 102(b) as being anticipated by Wu. Independent claim 28 presently recites (emphasis added):

28. A method, comprising:
performing a first allocation for a first amount of on-processor register space at the entry block of a routine, said first amount of on-processor register space for the use of said routine;

performing a second allocation for a second amount of on-processor register space for the storage of live information of said routine when said routine performs a function call to a second routine, said second amount less than said first amount;

performing said function call to said second routine; and,

performing a third allocation for a third amount of on-processor register space at the entry block of said second routine, said third amount of allocated on-processor register space for the use of said second routine and including a region of said first amount of allocated on-processor register space that stores stale information of said routine when said routine said performs said function call to said second routine.

The Applicants respectfully submit that independent claim 28 is patentable over Wu for the same reasons discussed above with respect to independent claim 28. That is, claim 28 recites a pair of allocations ("a first allocation" and "a second allocation") that allocate registers for the use of the same routine. Moreover, claim 28 recites that the later (second) allocation allocates for less register space than the earlier allocation.

These claimed features stem at least from the Applicants' discussion of Figures 5 and 6 of the present application. That is, Figures 5 and 6 of the present application and their surrounding discussion disclose a technique where an allocation instruction that allocates register space (e.g., space 610 in Figure 6) for a caller routine (as a consequence of the caller routine having a function call) can be configured to allocate for less register space than a previously executed allocation instruction allocated (e.g., space 605 of Figure 6) for the same caller routine (e.g., located at the caller routine's entry block).

By contrast the Wu reference at best only discloses, teaches or suggests an allocation instruction for the called routine rather than the caller routine. As such, the Wu reference fails to cover the Applicants' claimed subject matter. The Wu reference teaches the implementation of a software pipelined loop as a called sub function and the corresponding allocation of register space for the called sub function. See, Wu Col. 3,lines 26-29 and Col.4 lines 5-21. Nothing is said in Wu with respect to the insertion of an allocation instruction that allocates register space for the caller function, however. Therefore the Wu reference fails to anticipate each and every limitation of independent claim 28 of the present application.

Therefore independent claim 28 and its corresponding dependent claims are patentable over the Wu reference.

**Examiner's response:**

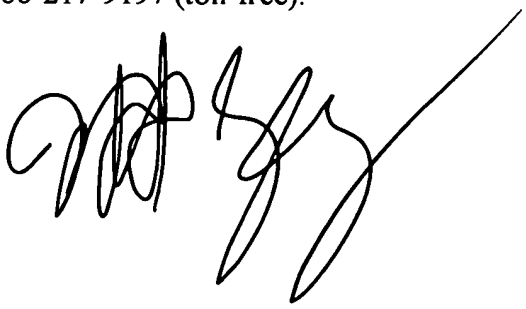
As in the argument for claim 1, the Applicant distinguishes between a caller routine and a called routine, but does not make that distinction in the claims. If this distinguishes the invention over the prior art, then that feature should be in the claims.

***Conclusion***

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (703) 305-8046. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703) 305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Lawrence Shrader
Examiner
Art Unit 2124

22 April 2004

TODD INGBERG
PRIMARY EXAMINER